CS210 PROJECT

Buket Özen 25441

Deniz Küçükahmetler 24879

Guljahan Annagurbanova 26278

Sabanur Mete 25473

Y. Dila Kurumahmutoğlu 25543

Introduction

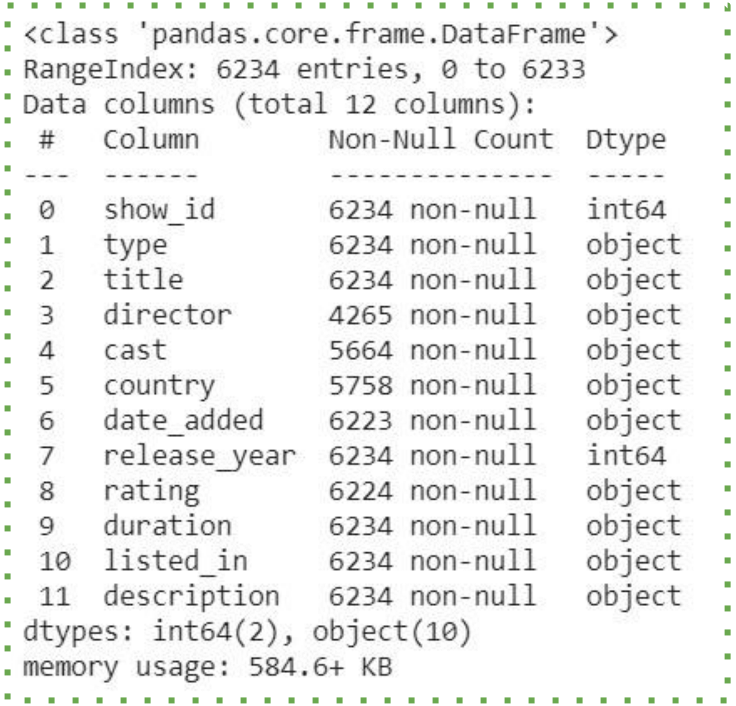
- Decision on topic
- The influence of the internet
- Guiding children for good

NETFLIX

Problem Definition

- Suitable contents for children audience
- Division of the data
- Correlations between attributes
- Child-friendliness relationship

Utilized Datasets: Info



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6234 entries, 0 to 6233
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       6234 non-null   int64
 1   type          6234 non-null   object
 2   title         6234 non-null   object
 3   director      4265 non-null   object
 4   cast          5664 non-null   object
 5   country       5758 non-null   object
 6   date_added    6223 non-null   object
 7   release_year  6234 non-null   int64
 8   rating        6224 non-null   object
 9   duration      6234 non-null   object
 10  listed_in     6234 non-null   object
 11  description   6234 non-null   object
dtypes: int64(2), object(10)
memory usage: 584.6+ KB
```

Information about the dataframe:

- 12 columns
- 6234 rows
- 2 integer valued columns
- 10 columns with dtype as object (string in pandas)

Utilized Datasets: Ratings

14 standardized abbreviations for movie and TV show ratings

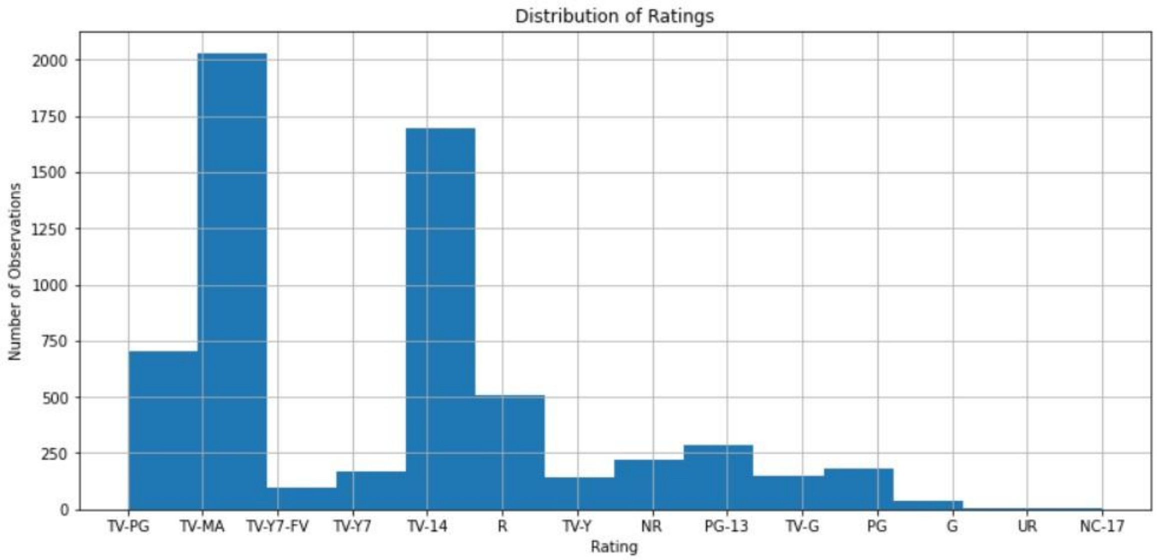
1. **G:** General
2. **TV-G:** General Audience
3. **PG:** Parental Guidance Recommended
4. **PG-13:** Parents Strongly Cautioned, Some Material May Be Inappropriate for Children Under 13
5. **TV-PG:** Parental Guidance Suggested
6. **TV-MA:** Mature Audience Only. Intended for adults and may be unsuitable for children under 17
7. **TV-Y7-FV:** Directed to Older Children - Fantasy Violence - Contains mild fantasy or comedic violence
8. **TV-Y7:** Directed to Older Children. Intended for children ages 7 and older
9. **TV-14:** Parents Strongly Cautioned. Intended for children ages 14 and older in the company of an adult
10. **R:** Restricted – Under 17 requires accompanying parent or adult guardian
11. **TV-Y:** All Children - programs aimed at a very young audience, including children from ages 2-6
12. **NR:** Not Rated
13. **UR:** Unrated
14. **NC-17:** No Children Under 17 Admitted

MATURE 17+
M
CONTENT RATED BY
ESRB

TV
Y

R
RESTRICTED
UNDER 17 REQUIRES ACCOMPANYING PARENT OR ADULT GUARDIAN

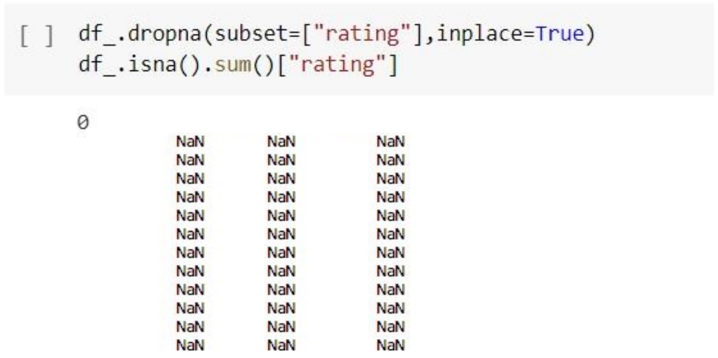
Utilized Datasets: Rating Distribution



```
rating
TV-MA       2027
TV-14       1698
TV-PG        701
R            508
PG-13        286
NR           218
PG           184
TV-Y7        169
TV-G         149
TV-Y         143
TV-Y7-FV      95
G             37
UR             7
NC-17          2
Name: show_id, dtype: int64
```

Utilized Datasets: Discarding Missing Values

A. NAN values in "rating" column is dropped.

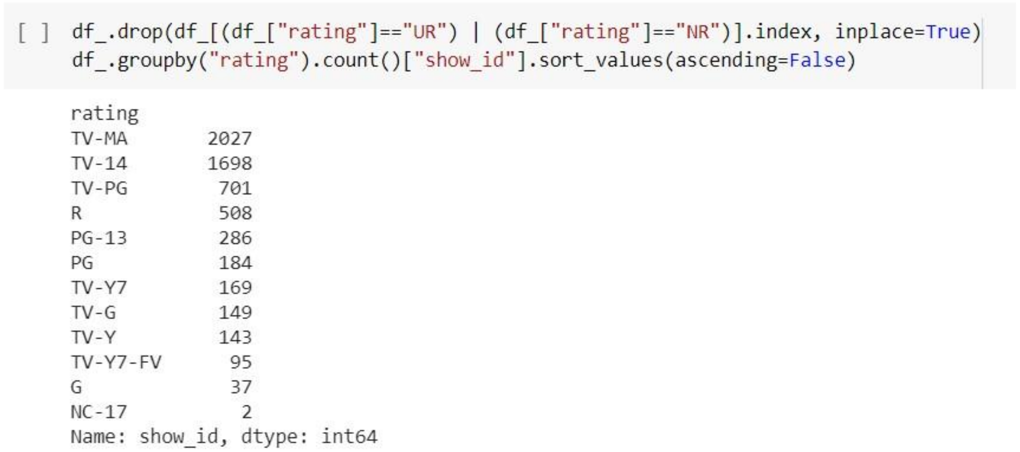


```
[ ]  df_.dropna(subset=["rating"],inplace=True)
     df_.isna().sum()["rating"]

0
```

NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN

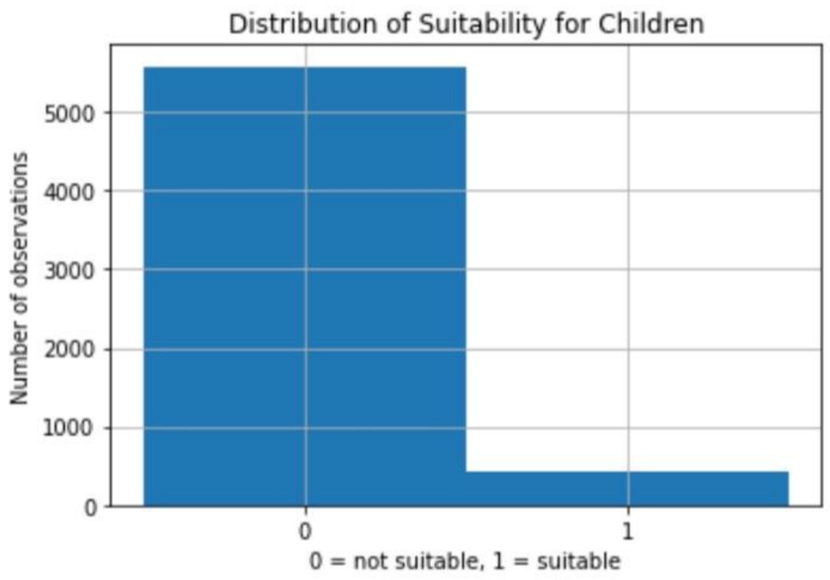
NAN values in important columns are converted into "Unknown".

B. the "UR" (unrated) and "NR" (not rated) rows are dropped.



```
[ ]  df_.drop(df_[(df_["rating"]=="UR") | (df_["rating"]=="NR")].index, inplace=True)
     df_.groupby("rating").count()["show_id"].sort_values(ascending=False)

rating
TV-MA       2027
TV-14       1698
TV-PG        701
R            508
PG-13        286
PG           184
TV-Y7        169
TV-G         149
TV-Y         143
TV-Y7-FV      95
G             37
NC-17          2
Name: show_id, dtype: int64
```

Utilized Datasets: Target Column



Children-friendly ratings = 1

- G
- TV_Y7
- TV-G
- TV-Y
- TV-Y7-FV

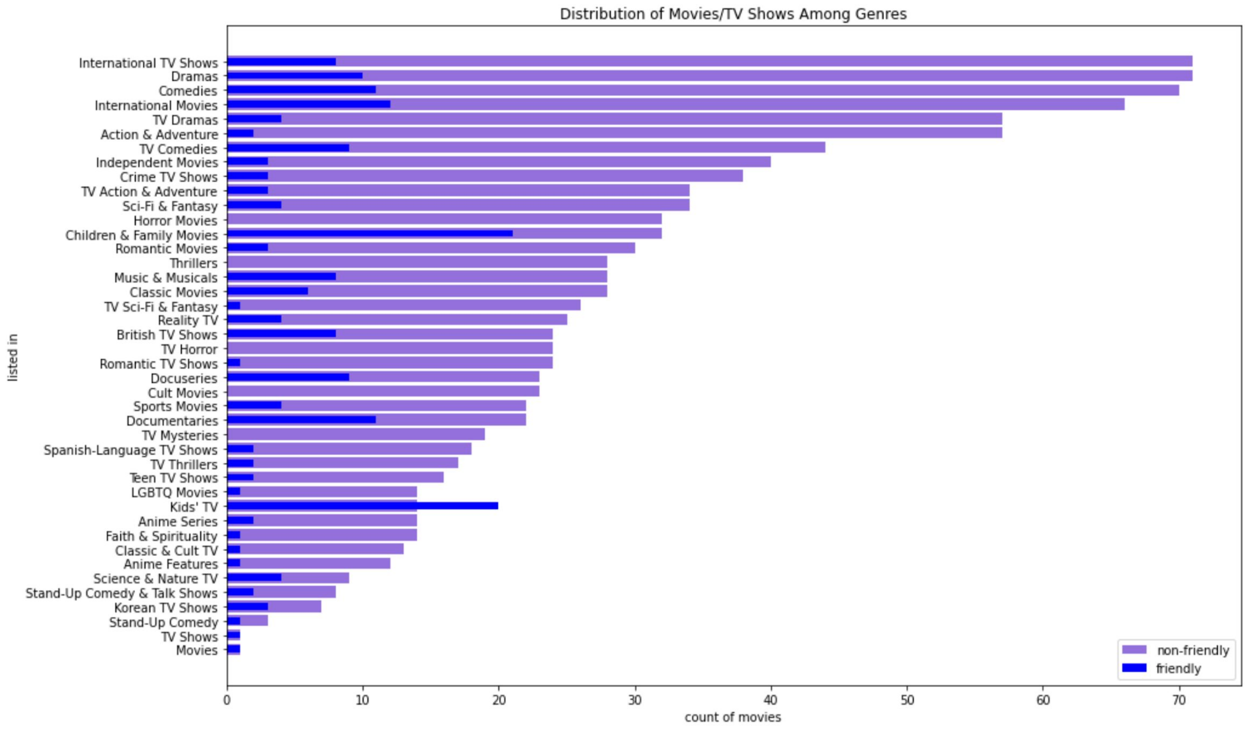
Not suitable for children = 0

Data Exploration

In data exploration, we looked at distributions of children-friendly and non-friendly content in tv-shows and movies and their possible correlations with other attributes.
We found out that there is a correlation between ***type*** of the content, ***director***,***genre***, ***cast***, ***duration***, ***description*** and ***suitability of the content for children audience***.

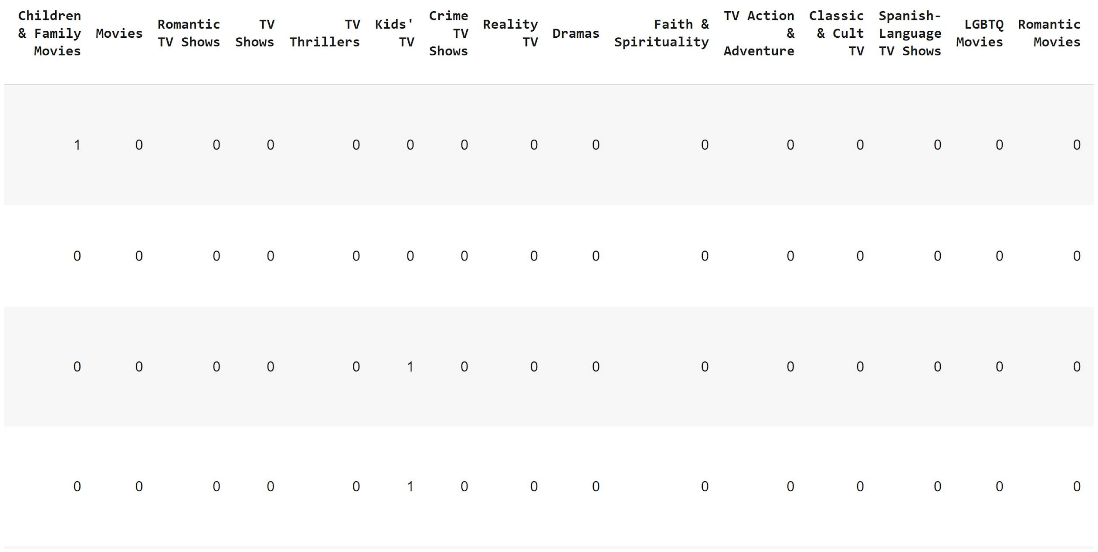
Data Exploration: Genres

41 different genres are found



Data Exploration: Genres

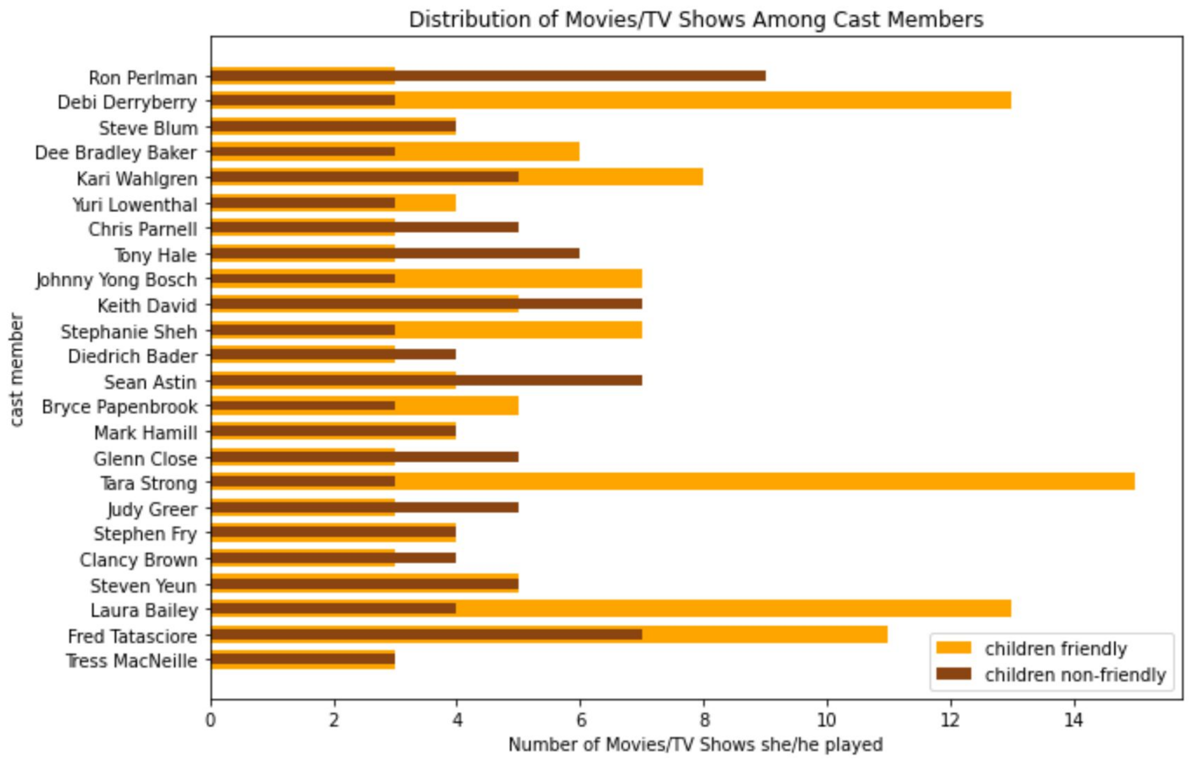
- **One-hot encoding** is used.
- Each movie/tv show's entries in those columns are marked as **1** if the movie/tv show includes that genre, otherwise it is marked as **0**.



Children & Family Movies	Movies	Romantic TV Shows	TV Shows	TV Thrillers	Kids' TV	Crime TV Shows	Reality TV	Dramas	Faith & Spirituality	TV Action & Adventure	Classic & Cult TV	Spanish-Language TV Shows	LGBTQ Movies	Romantic Movies
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

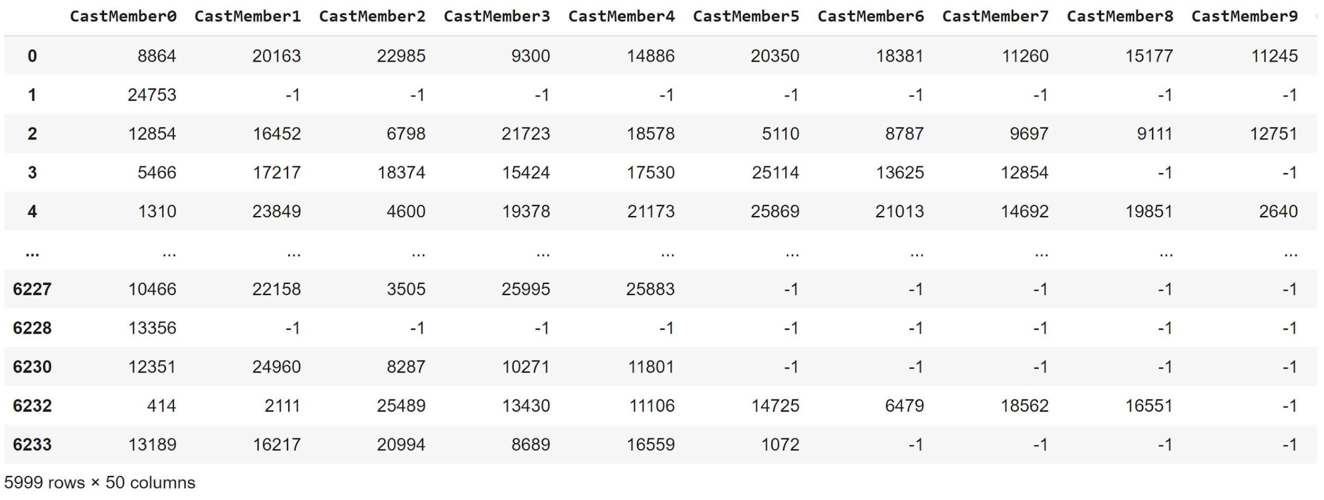
Data Exploration: Cast Members

26652 cast members found



Data Exploration: Cast Members

- **High number of cast members** -> couldn't use one-hot encoding
- Instead **cast members are indexed** and number of "highest number of cast members" columns are added.
- "-1" represents no cast member exists in that ranking.

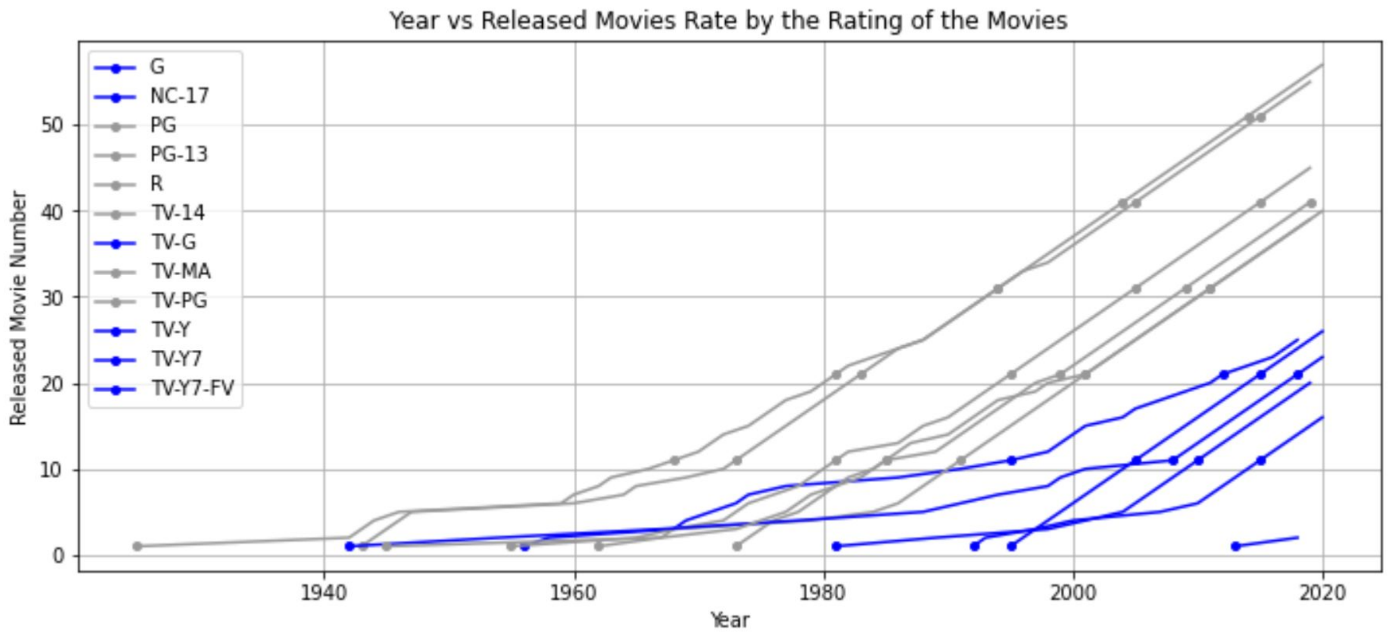


	CastMember0	CastMember1	CastMember2	CastMember3	CastMember4	CastMember5	CastMember6	CastMember7	CastMember8	CastMember9
0	8864	20163	22985	9300	14886	20350	18381	11260	15177	11245
1	24753	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	12854	16452	6798	21723	18578	5110	8787	9697	9111	12751
3	5466	17217	18374	15424	17530	25114	13625	12854	-1	-1
4	1310	23849	4600	19378	21173	25869	21013	14692	19851	2640
...	...	...	...	...	...	...	...	...	...	...
6227	10466	22158	3505	25995	25883	-1	-1	-1	-1	-1
6228	13356	-1	-1	-1	-1	-1	-1	-1	-1	-1
6230	12351	24960	8287	10271	11801	-1	-1	-1	-1	-1
6232	414	2111	25489	13430	11106	14725	6479	18562	16551	-1
6233	13189	16217	20994	8689	16559	1072	-1	-1	-1	-1

5999 rows × 50 columns

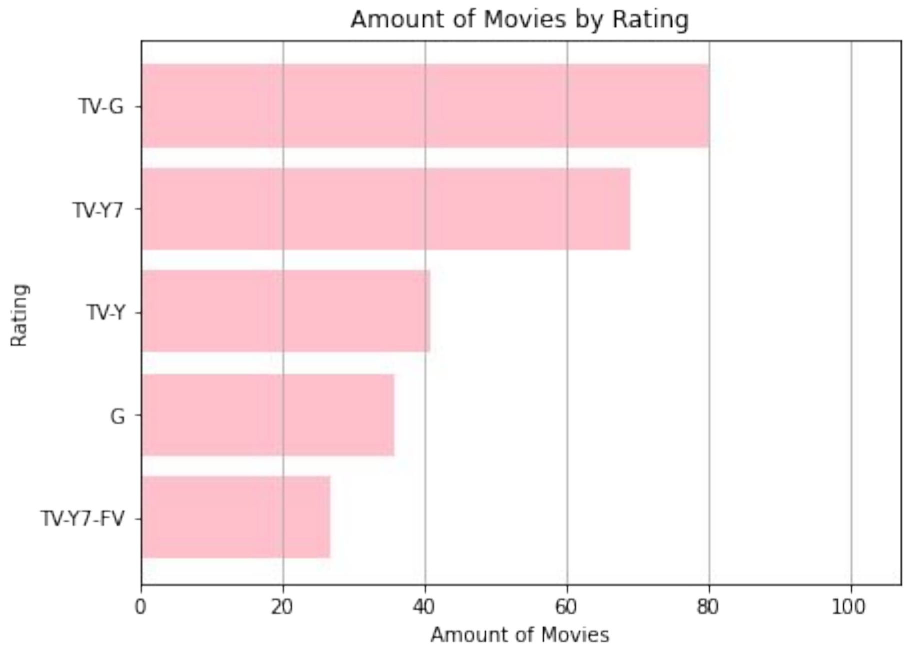
Data Exploration: Release Years

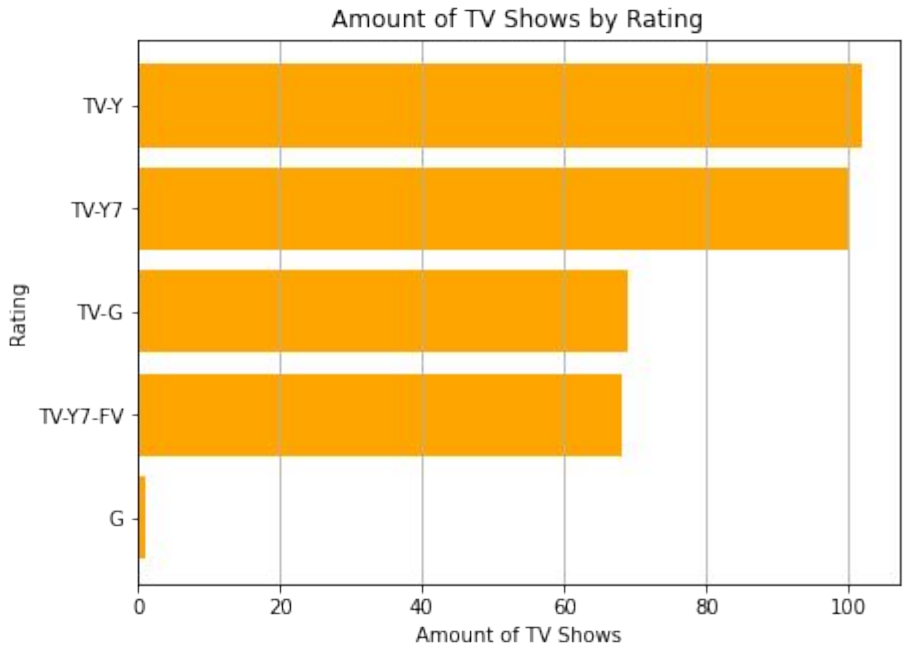
- No correlation found so we didn't consider the release year attribute for machine learning models.



Data Exploration: Types

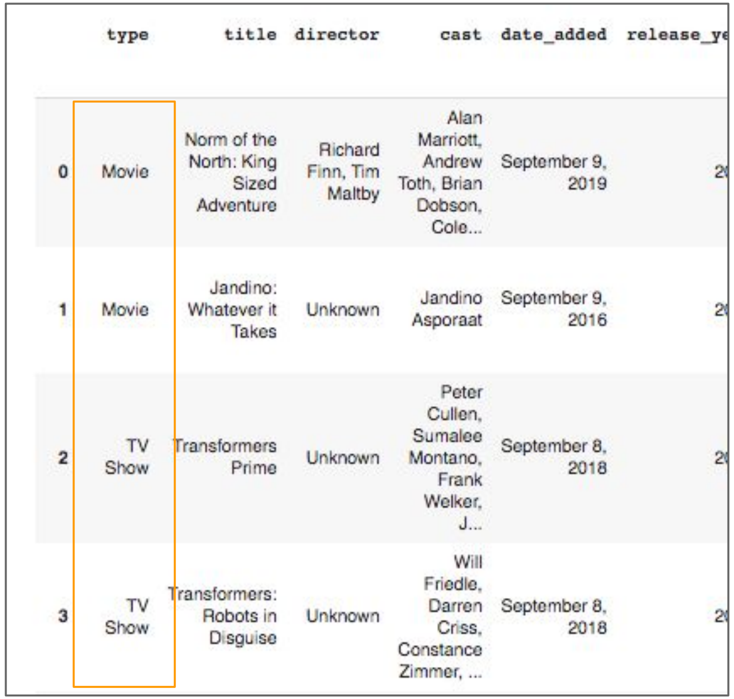
Distribution of ratings suitable for children in movies and tv shows



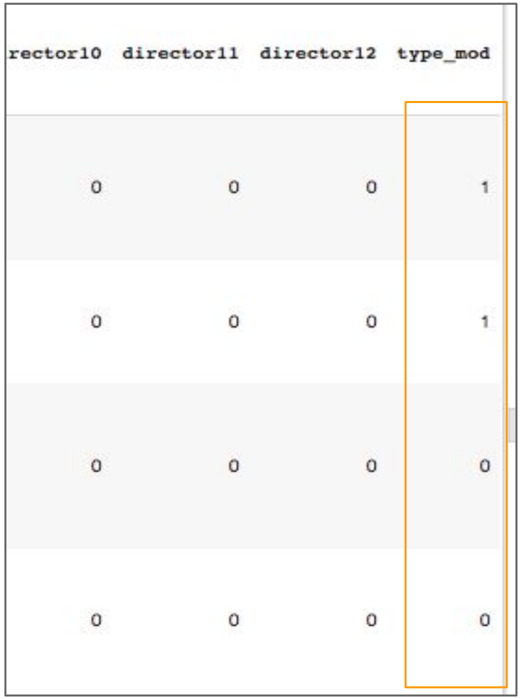


Data Exploration: Types

Due to the correlation we have observed between type of content and its suitability for children, we have added type_mod column to our model. Movies got value of 1 and tv-shows 0.

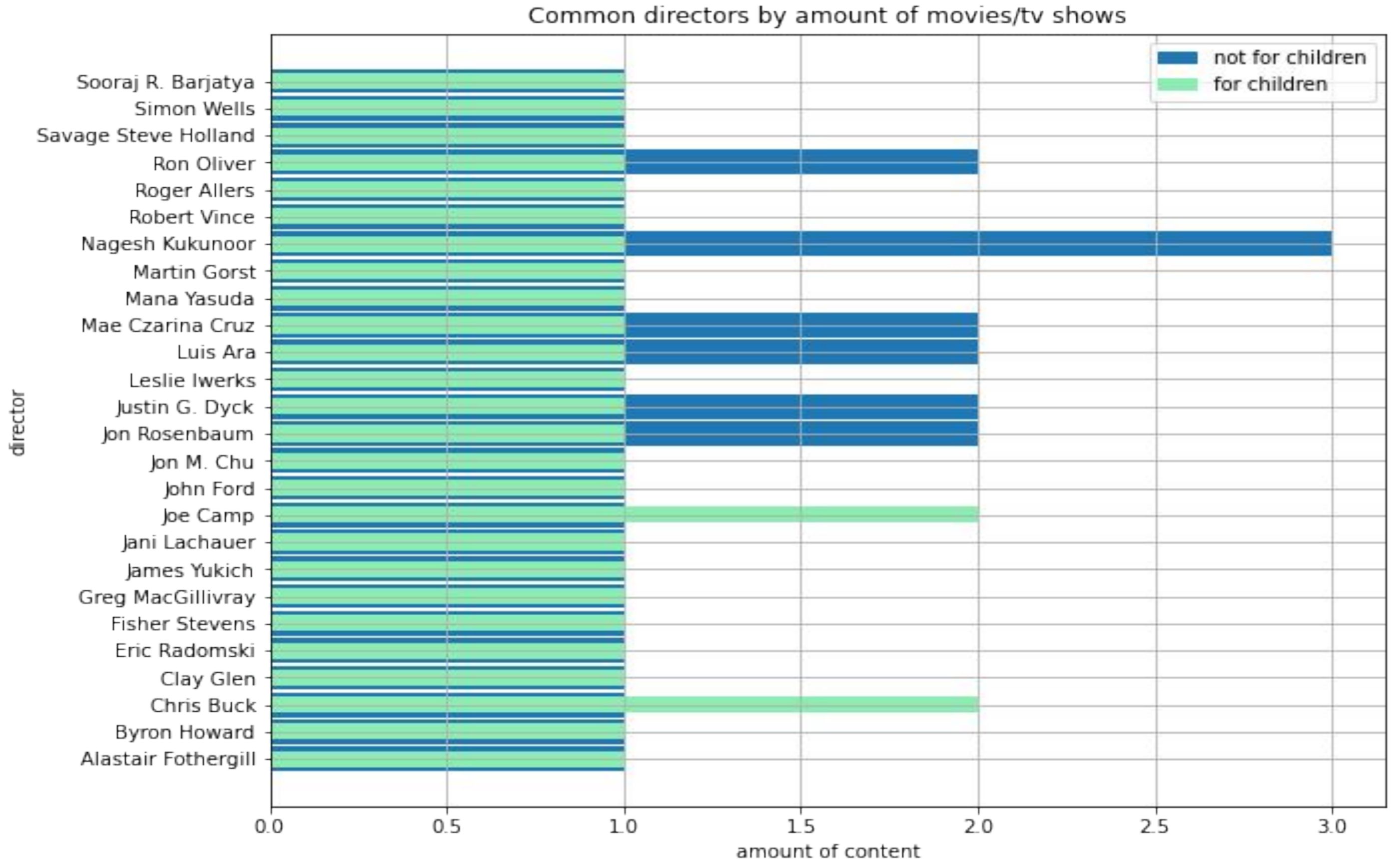


	type	title	director	cast	date_added	release_ye
0	Movie	Norm of the North: King Sized Adventure	Richard Finn, Tim Maltby	Alan Marriott, Andrew Toth, Brian Dobson, Cole...	September 9, 2019	2
1	Movie	Jandino: Whatever it Takes	Unknown	Jandino Asporaat	September 9, 2016	2
2	TV Show	Transformers Prime	Unknown	Peter Cullen, Sumalee Montano, Frank Welker, J...	September 8, 2018	2
3	TV Show	Transformers: Robots in Disguise	Unknown	Will Friedle, Darren Criss, Constance Zimmer, ...	September 8, 2018	2



rector10	director11	director12	type_mod
0	0	0	1
0	0	0	1
0	0	0	0
0	0	0	0

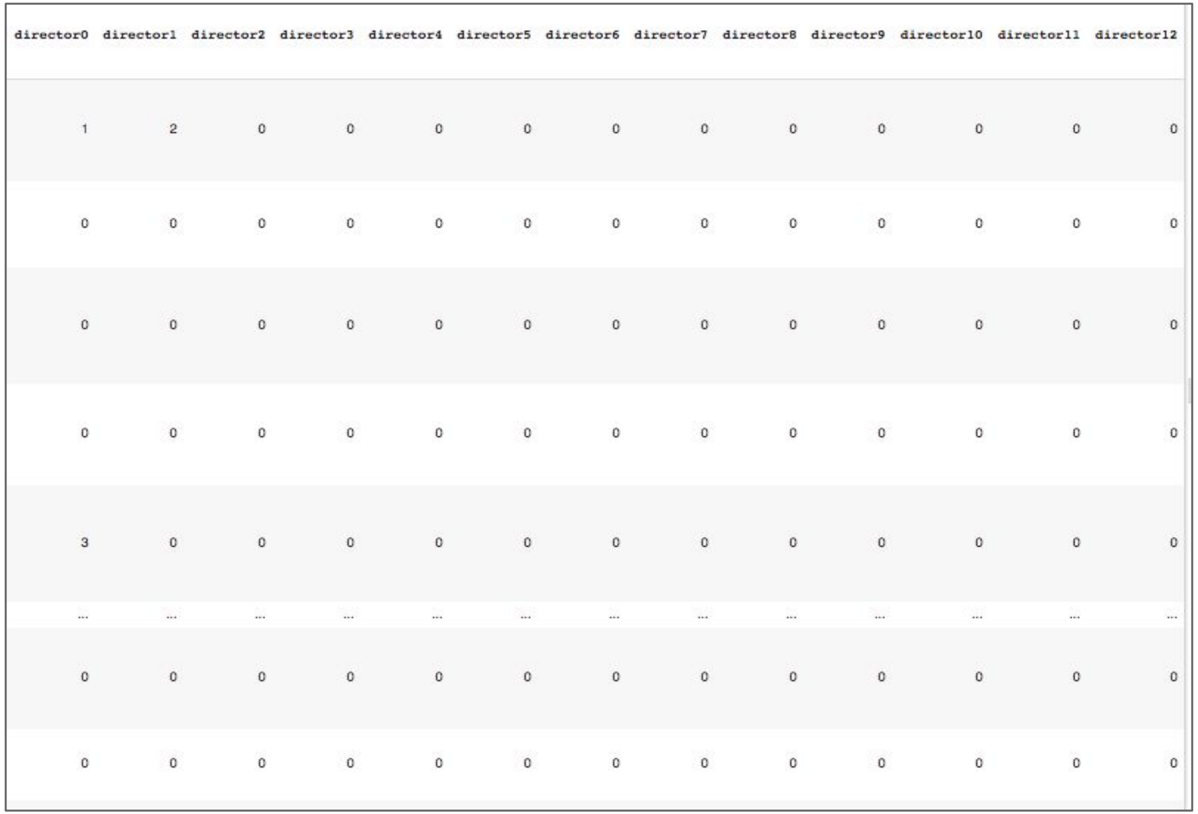
Data Exploration: Directors



Total number of directors: 3475

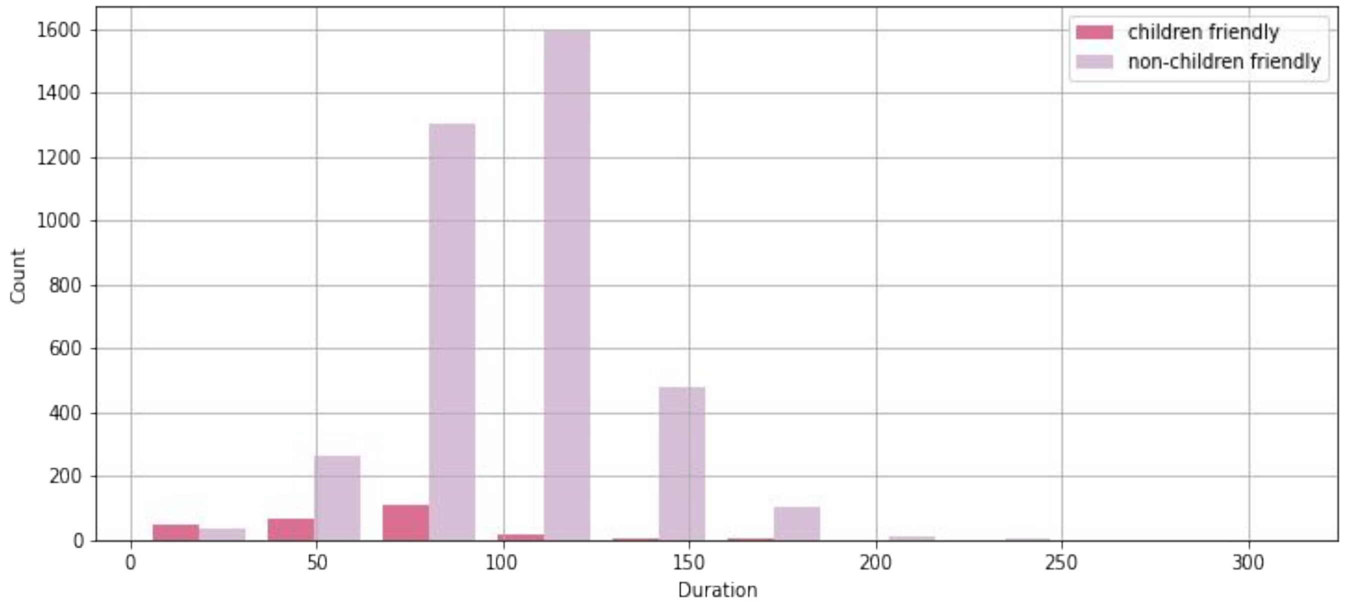
Data Exploration: Directors

- Didn’t use one-hot-encoding method because the total number of directors is more than 3000.
- found the maximum number of directors (max_dir) we could have in a movie/tv show,
- added max_dir many columns to our extended dataframe, each director had a unique id(starting from 1);
- then filled our ext_df extended dataframe with director ids for each movie/show



director0	director1	director2	director3	director4	director5	director6	director7	director8	director9	director10	director11	director12
1	2	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Data Exploration: Duration

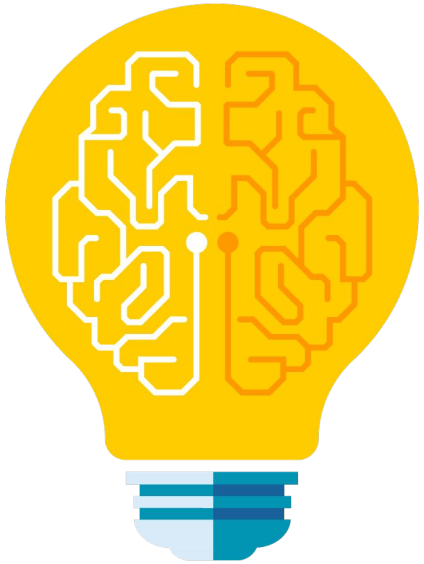


Children-friendly movies:
Max - Min duration: 180 - 3.
Mean duration: ~ 64.165.

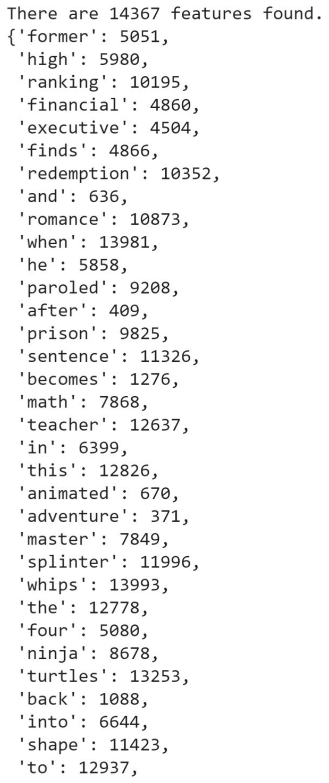
Children-unfriendly movies:
Max - Min duration: 312 - 12.
Mean duration: ~ 101.719

Machine Learning Models

- Random Forest
- SVM



Consideration of the “description” Attribute

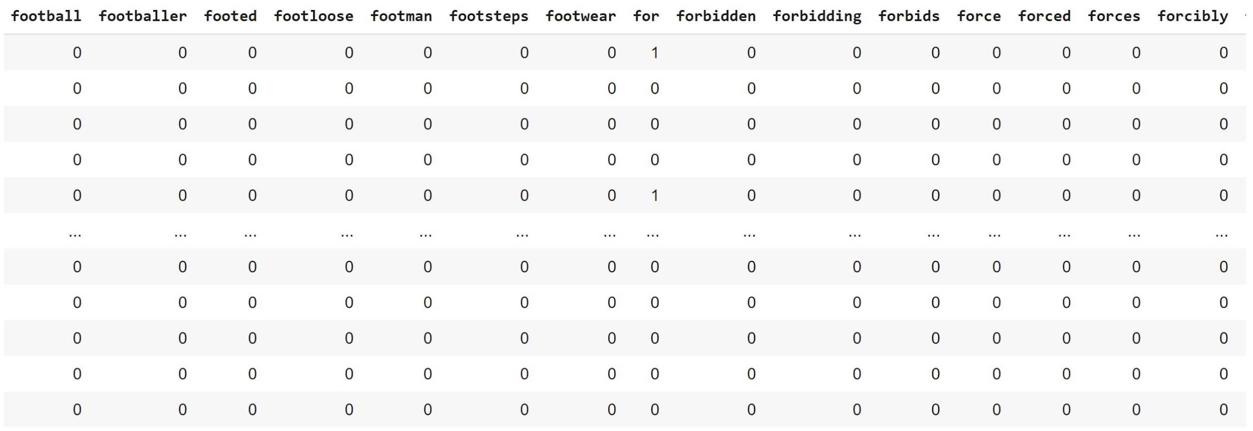


```
There are 14367 features found.
{'former': 5051,
 'high': 5980,
 'ranking': 10195,
 'financial': 4860,
 'executive': 4504,
 'finds': 4866,
 'redemption': 10352,
 'and': 636,
 'romance': 10873,
 'when': 13981,
 'he': 5858,
 'paroled': 9208,
 'after': 409,
 'prison': 9825,
 'sentence': 11326,
 'becomes': 1276,
 'math': 7868,
 'teacher': 12637,
 'in': 6399,
 'this': 12826,
 'animated': 670,
 'adventure': 371,
 'master': 7849,
 'splinter': 11996,
 'whips': 13993,
 'the': 12778,
 'four': 5080,
 'ninja': 8678,
 'turtles': 13253,
 'back': 1088,
 'into': 6644,
 'shape': 11423,
 'to': 12937,
```

- “Bag-of-words” is used
- All words used in the description are vectorized with their counts.
- Stop-words are removed.

Concatenating Word Counts to the Data Frame

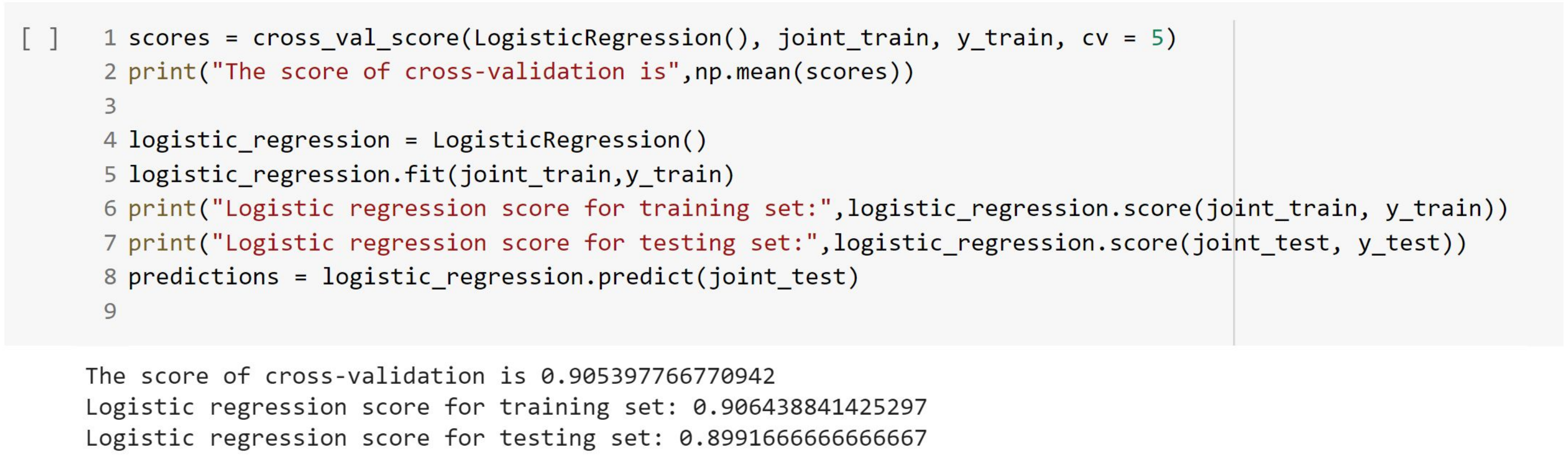
- After seperation of the data frame to test and train datasets, word counts of each movie/tv show is concatenated with the data frame
- 14344 columns in total



football	footballer	footed	footloose	footman	footsteps	footwear	for	forbidden	forbidding	forbids	force	forced	forces	forcibly
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Logistic Regression

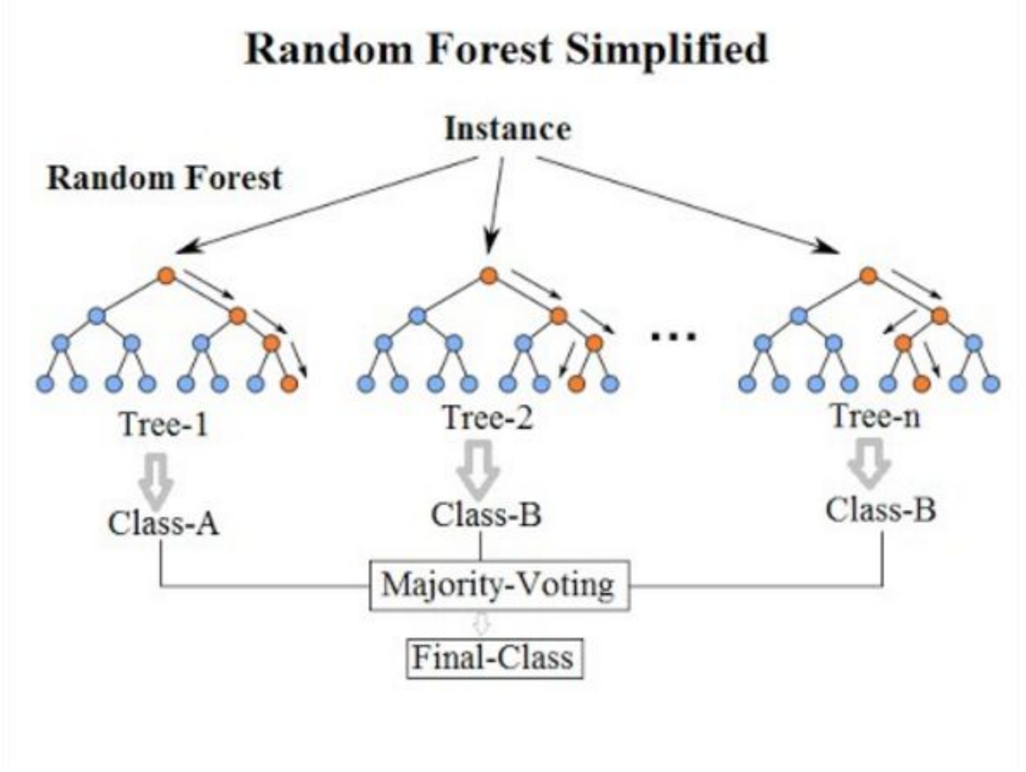
- Cross validation and logistic regression is performed



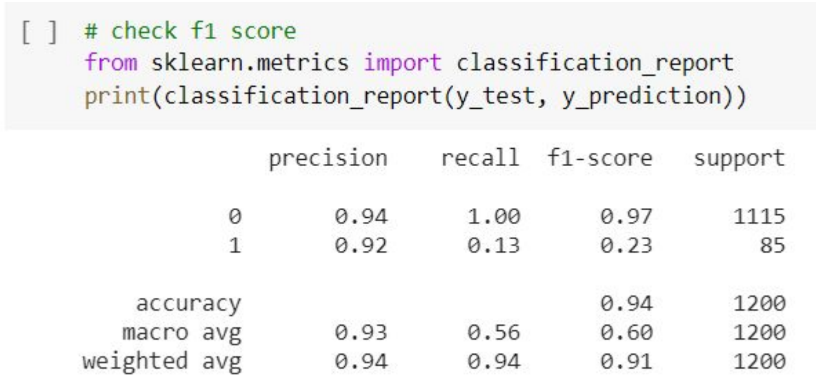
```
[ ]
scores = cross_val_score(LogisticRegression(), joint_train, y_train, cv = 5)
print("The score of cross-validation is",np.mean(scores))

logistic_regression = LogisticRegression()
logistic_regression.fit(joint_train,y_train)
print("Logistic regression score for training set:",logistic_regression.score(joint_train, y_train))
print("Logistic regression score for testing set:",logistic_regression.score(joint_test, y_test))
predictions = logistic_regression.predict(joint_test)

```

```
The score of cross-validation is 0.905397766770942
Logistic regression score for training set: 0.906438841425297
Logistic regression score for testing set: 0.8991666666666667
```

Random Forest Model



- Initial accuracy: 93.66 %
- Accuracy value after grid search and confusion matrix: 93.75%



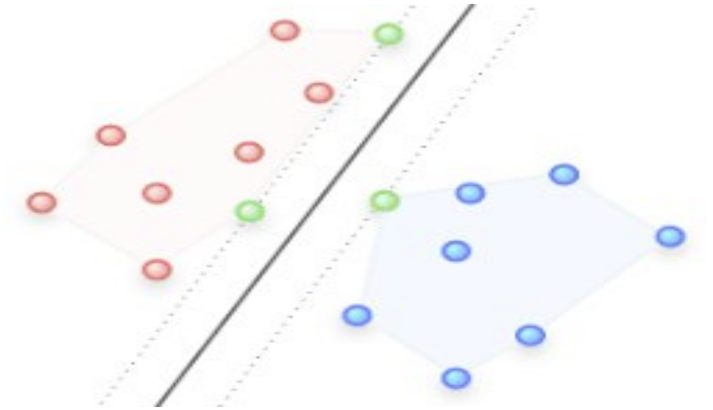
```
[ ]  # check f1 score
     from sklearn.metrics import classification_report
     print(classification_report(y_test, y_prediction))
```

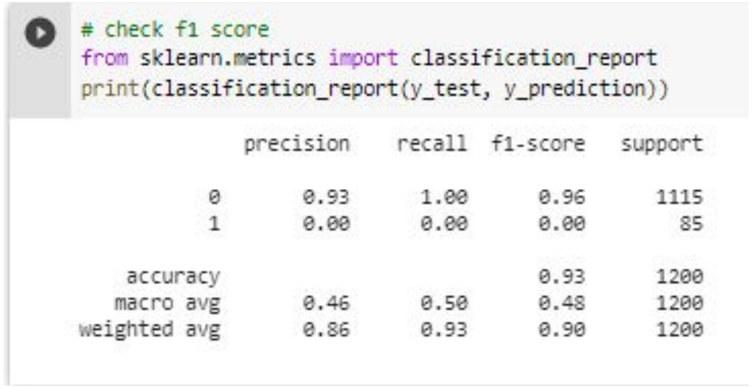
```
              precision    recall  f1-score   support

           0       0.94      1.00      0.97      1115
           1       0.92      0.13      0.23        85

    accuracy                           0.94      1200
   macro avg       0.93      0.56      0.60      1200
weighted avg       0.94      0.94      0.91      1200
```

Wikipedia. “Random Forest”. Last Modified at January 6, 2021.
https://en.wikipedia.org/wiki/Random_forest

SVM

- Accuracy score of SVM = 0.92916
- Accuracy after grid search & confusion matrix = 0.92916
- MCC score





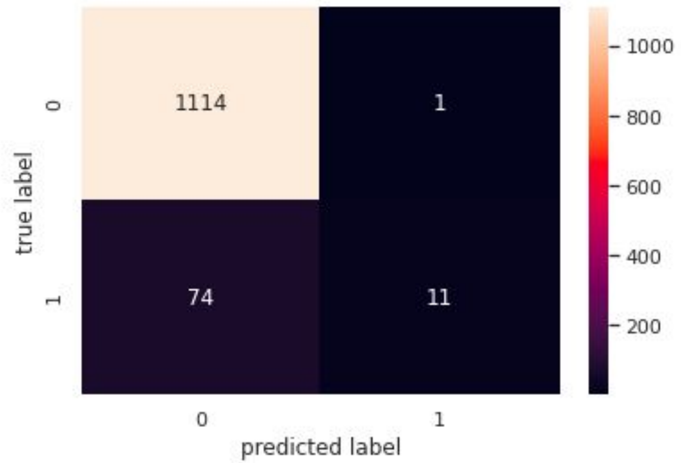
```
# check f1 score
from sklearn.metrics import classification_report
print(classification_report(y_test, y_prediction))
```

```
              precision    recall  f1-score   support

           0       0.93      1.00      0.96      1115
           1       0.00      0.00      0.00        85

    accuracy                           0.93      1200
   macro avg       0.46      0.50      0.48      1200
weighted avg       0.86      0.93      0.90      1200
```

Results and Discussion

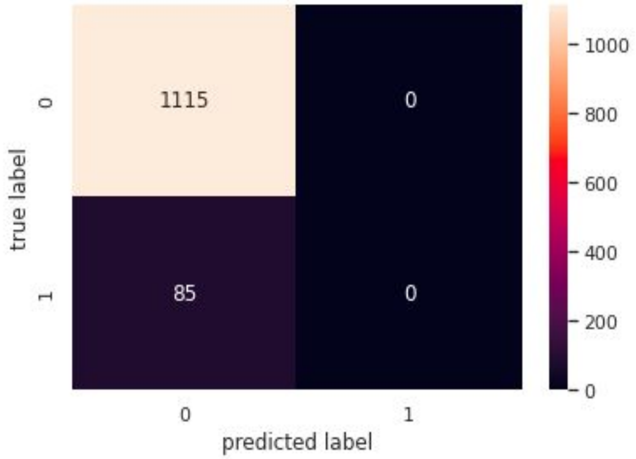
The confusion matrix obtained from the first model

* Possible predicted classes: 1 and 0.
* The classifier made a total of 1200 predictions
* Out of 1200 cases, the classifier predicted 1 -> 12 times,0 -> 1188 times.
* In reality, we have 85 1's and 1115 0's.
* At the end, this model classified 1125 data points correctly



The confusion matrix obtained from the second model

- Possible predicted classes: 1 and 0.
- The classifier made a total of 1200 predictions
- Out of 1200 cases, the classifier predicted 1 -> 0 times,0 -> 1200 times.
- In reality, we have 85 1's and 1115 0's.
- At the end, this model clas

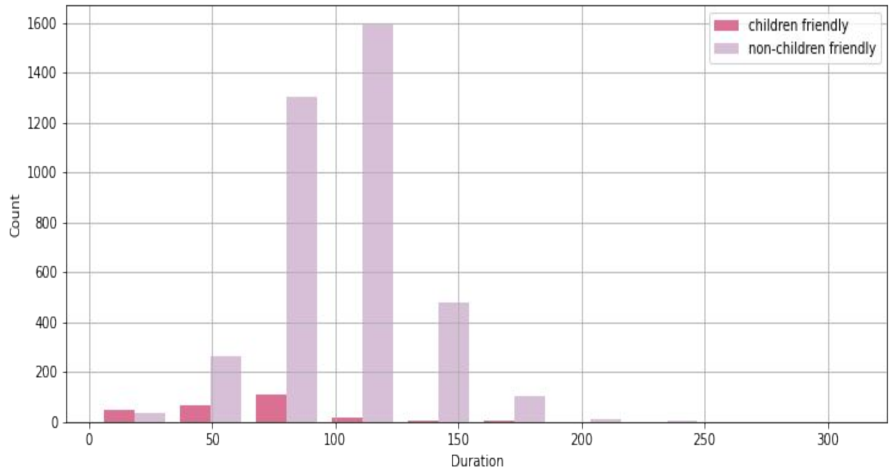


Conclusion

- What have we done?
- How did we do?
- Is our solution applicable?
- Advantages/disadvantages

Future Work

- The duration column could be added to the machine learning model, since there is a significant correlation between duration and suitability of the content for children. To achieve that, the duration of tv shows could be extracted in “minutes” since they are represented in “seasons” in the current dataset.



- Moreover, description of the movies/tv shows can be evaluated using deep learning algorithms to take their meanings into account rather than only considering the word counts as we implemented in the project.

DEEP
LEARNING